# AWS-Load-balancer

Launch EC2 Instance (Web Server – 1)

Main Task : **Distributing Traffic Between Two Apache Web Servers Using AWS Application Load Balancer**

Task 1: **Create EC2 Instance**

Logged in to AWS Management Console

Navigated to EC2 → Instances → Launch Instance

Instance Name: web-server-1

AMI: Amazon Linux 2

Instance Type: t2.micro

Key Pair: Selected existing key pair

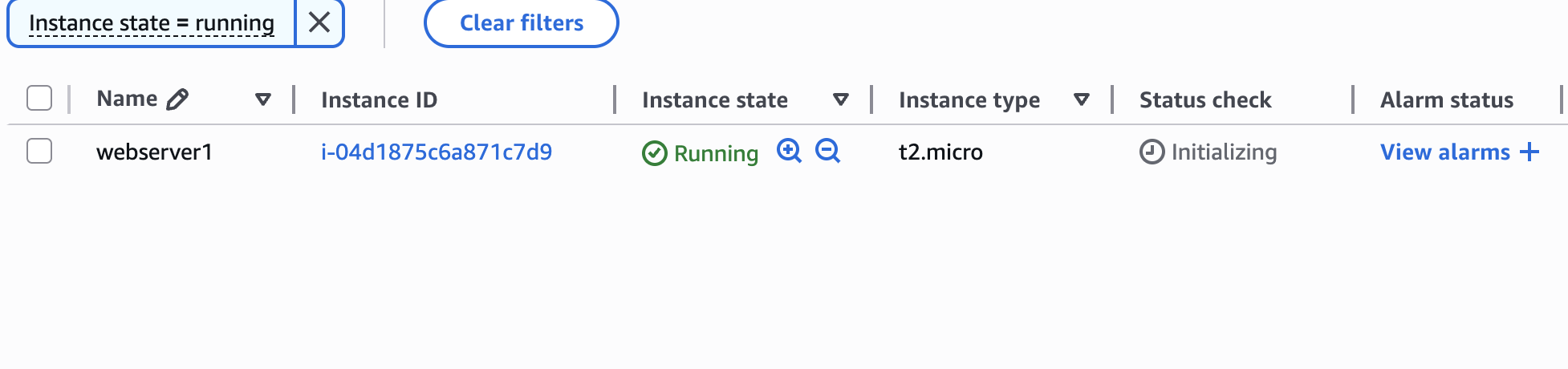
**Security Group Configuration**

Created a new security group with the following inbound rules:

Type Protocol Port Source

HTTP TCP 80 Anywhere (0.0.0.0/0) (Reason: HTTP access is required for web )

SSH TCP 22 (default)



Launch Instance

Task 2: **Launch Second EC2 Instance (Web Server – 2)**

## **Create EC2 Instance**

* Navigated to **EC2 → Instances → Launch Instance**
* Instance Name: **web-server-2**
* AMI: **Amazon Linux 2**
* Instance Type: **t2.micro**
* Key Pair: Selected existing key pair

Configure Network Settings

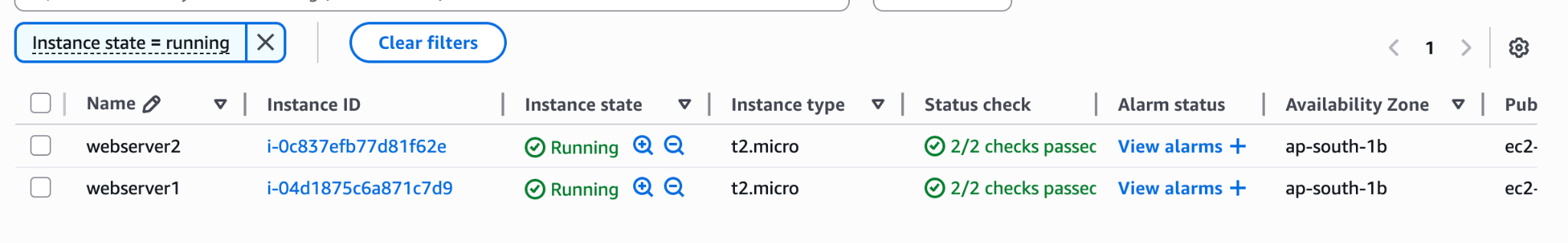
## **Security Group Configuration**

Created a new security group with the following inbound rules:

Type Protocol Port Source

HTTP TCP 80 Anywhere (0.0.0.0/0) (Reason: HTTP access is required for web )

SSH TCP 22 (default)



Task 3: **Connecting EC2 Instances to Local Terminal –webserver1**

Steps: terminal

>>Navigate to Downloads Folder

cd ~/Downloads

>>Set Key File Permissions

ls

>>Connect to web-server-1

ssh -i my-key.pem ec2-user@<web-server-1-public-ip>

>>Switch to Root User

Sudo -i

Task 4: To install Apache HTTP Server

sudo yum install httpd -y

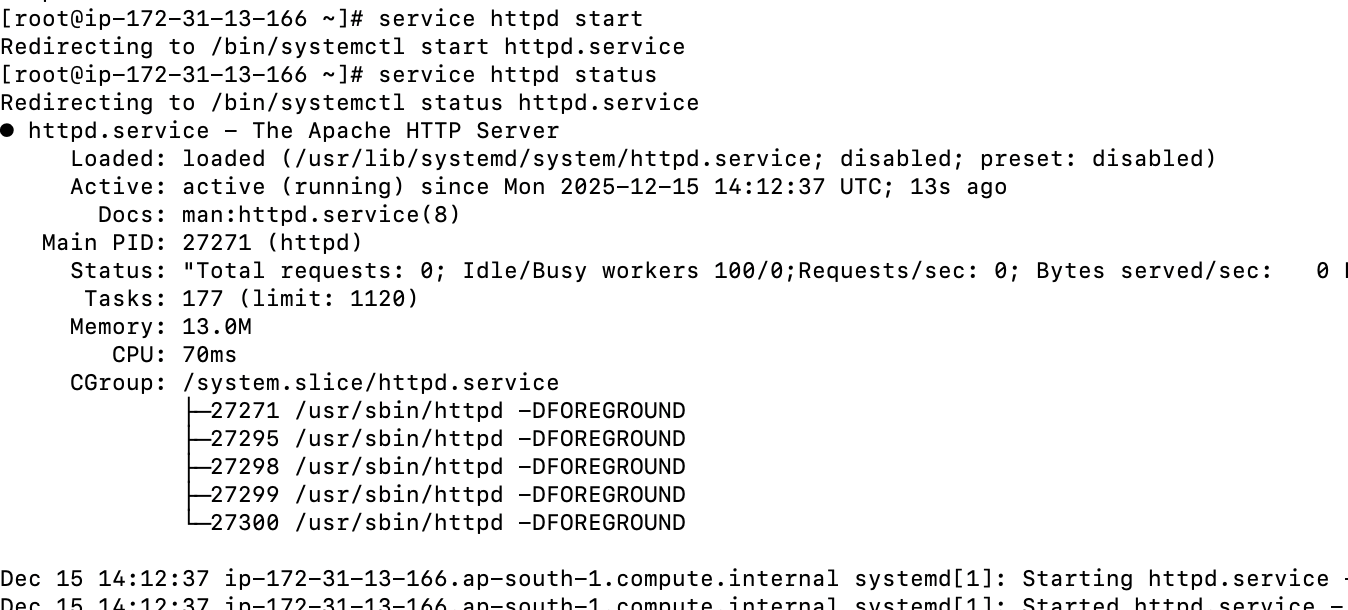
>>To start the web server

sudo systemctl start httpd

sudo systemctl enable httpd

>>check the status

systemctl status httpd



To exit from this press ctrl+c

>>once exit from this status it will come to ec2 user again change the user to root user

Cmd : sudo -i

Task 5: Create HTML File

>>Navigate to Web Directory

cd /var/www/html

>>Create HTML File

vi index.html

**Sample code**

<!DOCTYPE html>

<html>

<head>

<title>AWS Load Balancer Demo</title>

<style>

body {

margin: 0;

font-family: Arial, Helvetica, sans-serif;

background: linear-gradient(to right, #141e30, #243b55);

color: white;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.card {

background: #1f2a44;

padding: 30px;

border-radius: 12px;

text-align: center;

box-shadow: 0 10px 25px rgba(0,0,0,0.4);

width: 400px;

}

h1 {

margin-bottom: 10px;

color: #4fd1c5;

}

p {

font-size: 16px;

margin: 8px 0;

}

.server {

font-size: 18px;

font-weight: bold;

color: #f6e05e;

}

</style>

</head>

<body>

<div class="card">

<h1>AWS Load Balancer</h1>

<p>Web Application is Running Successfully 🚀</p>

<p class="server">Server Name: WEB-SERVER-1</p>

<p>Application Load Balancer Demo</p>

</div>

</body>

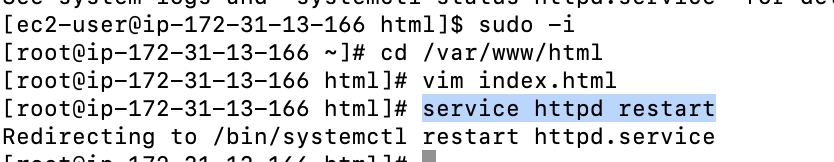
</html>

>>Exit Editor

:q!

>> restart the service

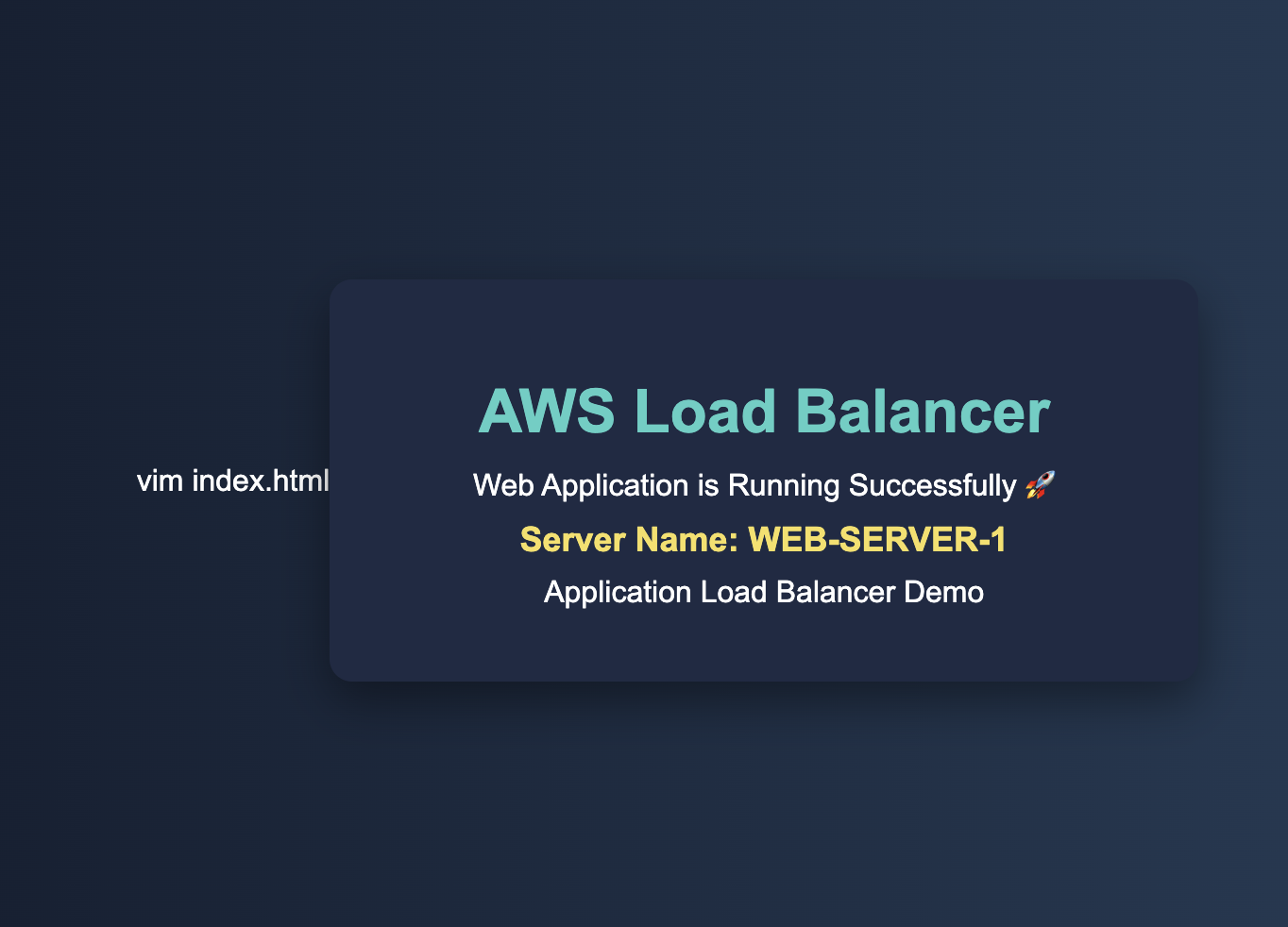
service httpd restart



>>go to chrome copy the public ip address and add http port

Ex: 13.203.75.16:80

Olp :web page



Task 6: **Connecting EC2 Instances to Local Terminal –webserver2**

>>repeat the same steps which we did in webserver1

>>Switched to root user.

>>Installed Apache web server.

sudo yum install httpd -y

>>Started and enabled Apache service.

sudo systemctl start httpd

sudo systemctl enable httpd

>>Verified Apache service status.

systemctl status httpd

Task 7: Created an HTML file.

cd /var/www/html

vi index.html

**Sample code for web server -2**

<!DOCTYPE html>

<html>

<head>

<title>AWS Load Balancer - Server 2</title>

<style>

body {

margin: 0;

font-family: Arial, sans-serif;

background: linear-gradient(to right, #0f2027, #203a43, #2c5364);

color: #ffffff;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.box {

background: rgba(0, 0, 0, 0.6);

padding: 35px;

border-radius: 14px;

text-align: center;

box-shadow: 0 12px 30px rgba(0,0,0,0.5);

width: 420px;

}

h1 {

color: #ffb703;

margin-bottom: 12px;

}

p {

font-size: 16px;

margin: 8px 0;

}

.server {

font-size: 20px;

font-weight: bold;

color: #90dbf4;

}

</style>

</head>

<body>

<div class="box">

<h1>Application Load Balancer</h1>

<p>Web Application is running successfully ✅</p>

<p class="server">Server Name: WEB-SERVER-2</p>

<p>Traffic served by EC2 Instance 2</p>

</div>

</body>

</html>

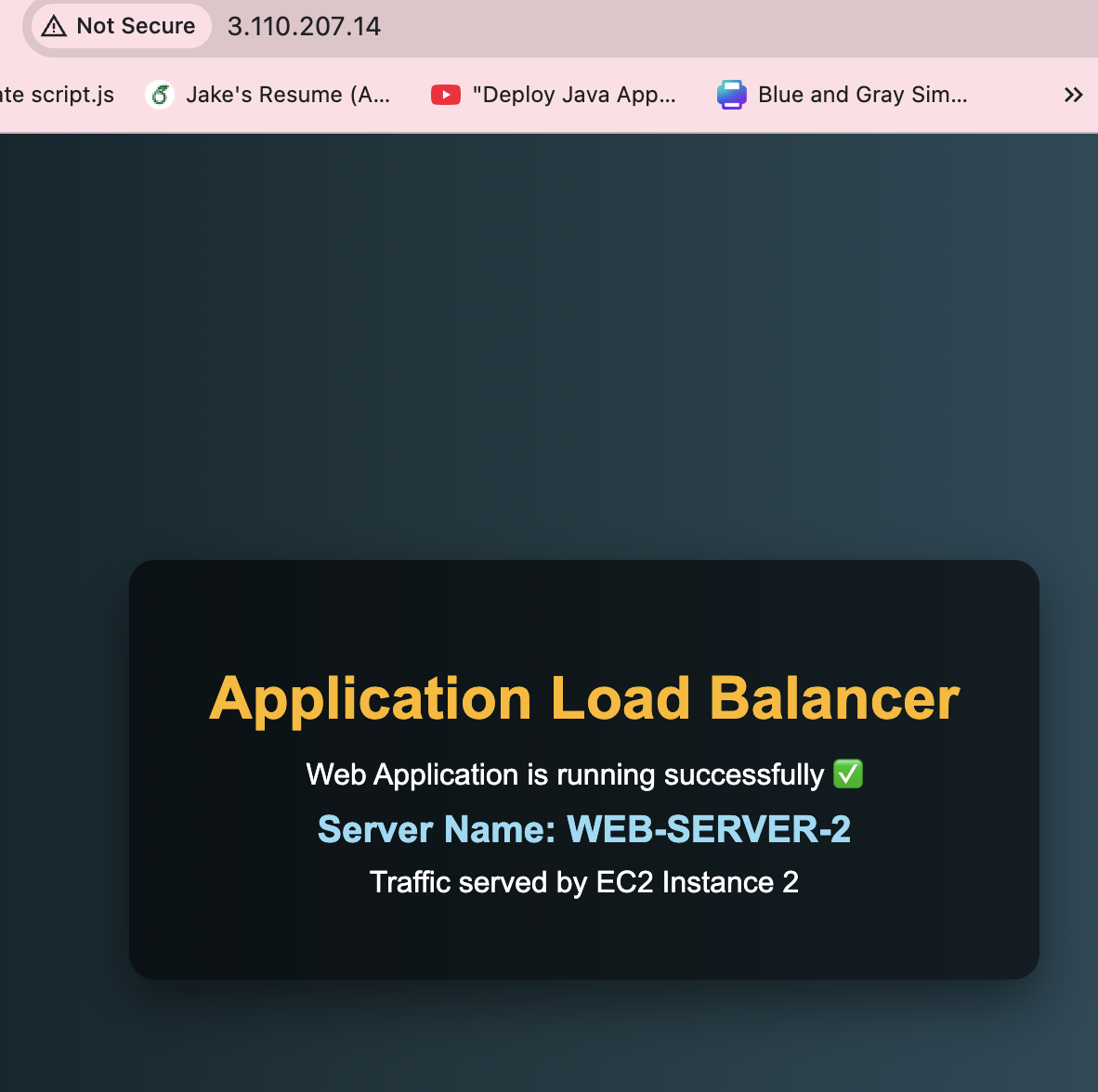
>>Exit Editor

:q!

>>go to chrome copy the public ip address and add http port

Ex: 3.110.207.14:80

Olp :web page



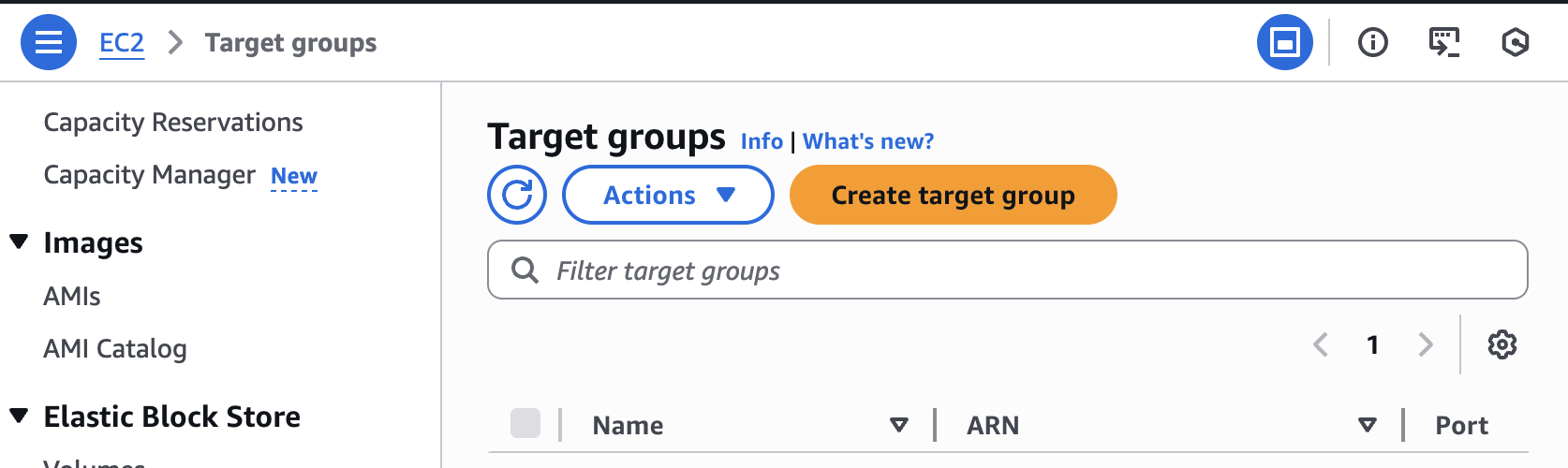
**Info :** In real time, we don’t share server IP addresses because it is not safe.

use a load balancer to share traffic between servers so the application works fast

Task 8**:Create Target group**

**Info :** A target group connects the load balancer to the web servers so traffic goes to the right servers**.**

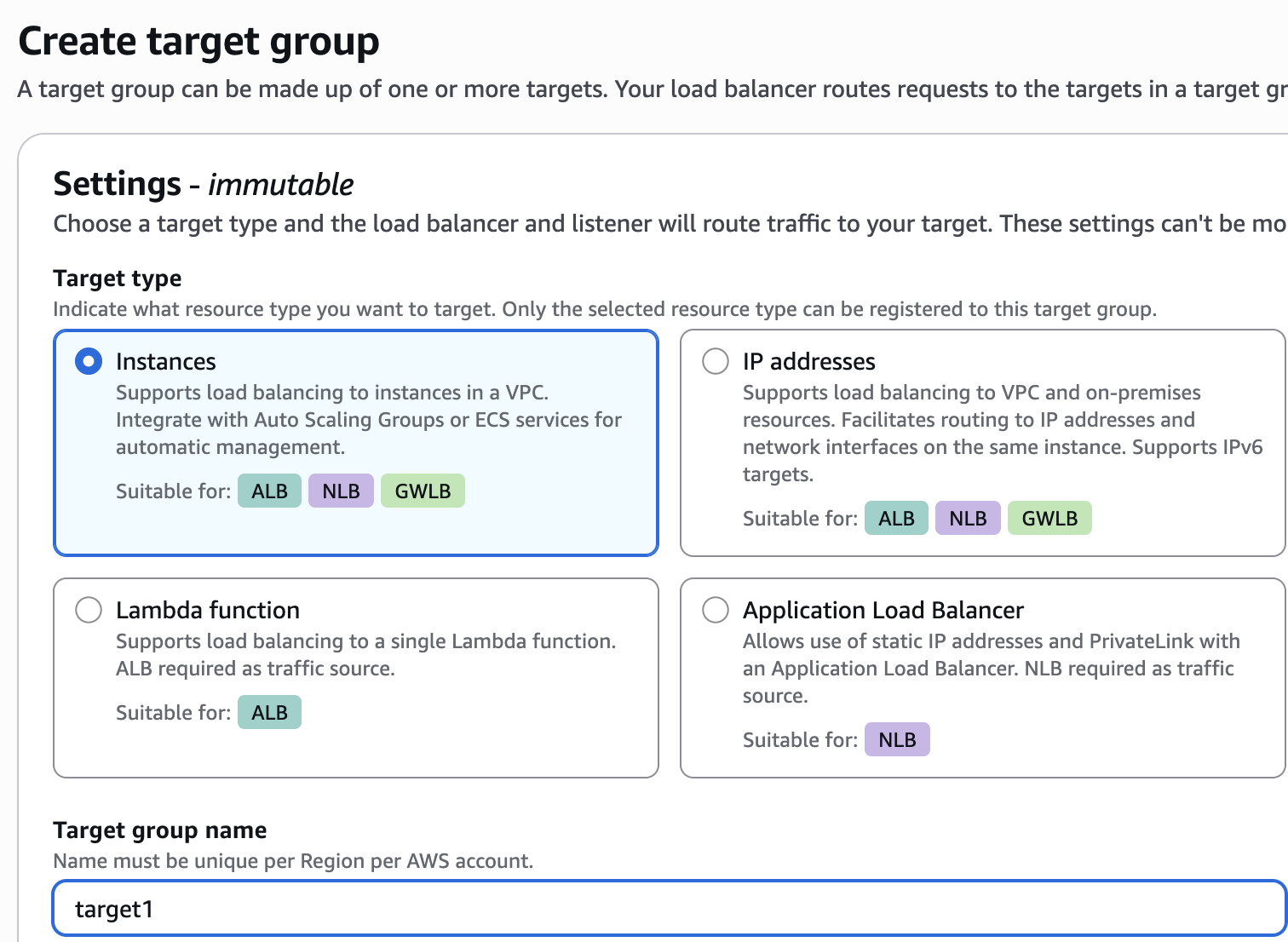
**Process:** Go to EC2 → Target Groups.

****

**>> configure the Target group**

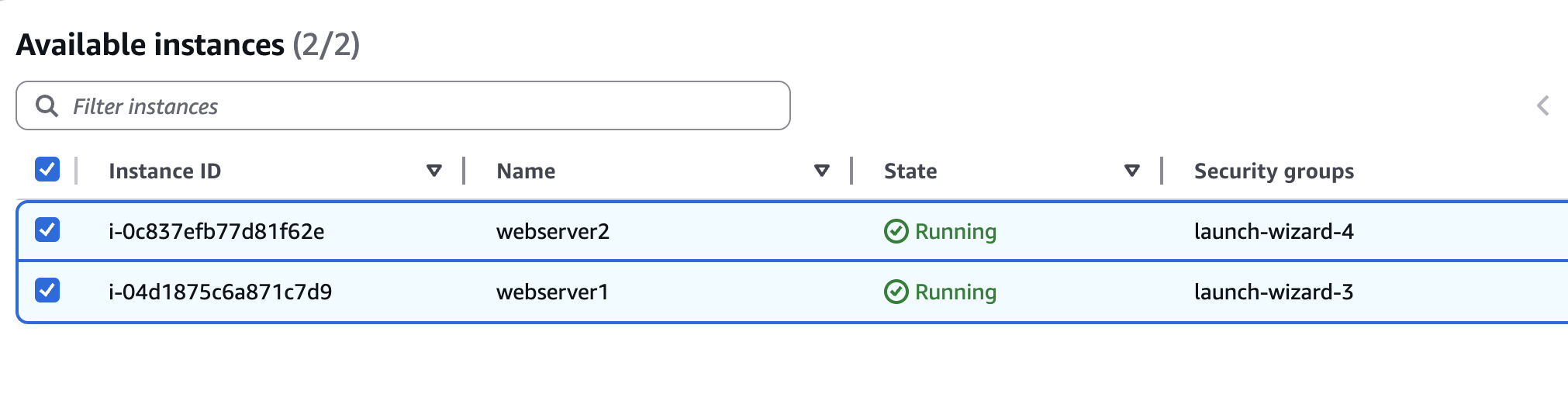
Enter the target name: target 1

Target type : instances

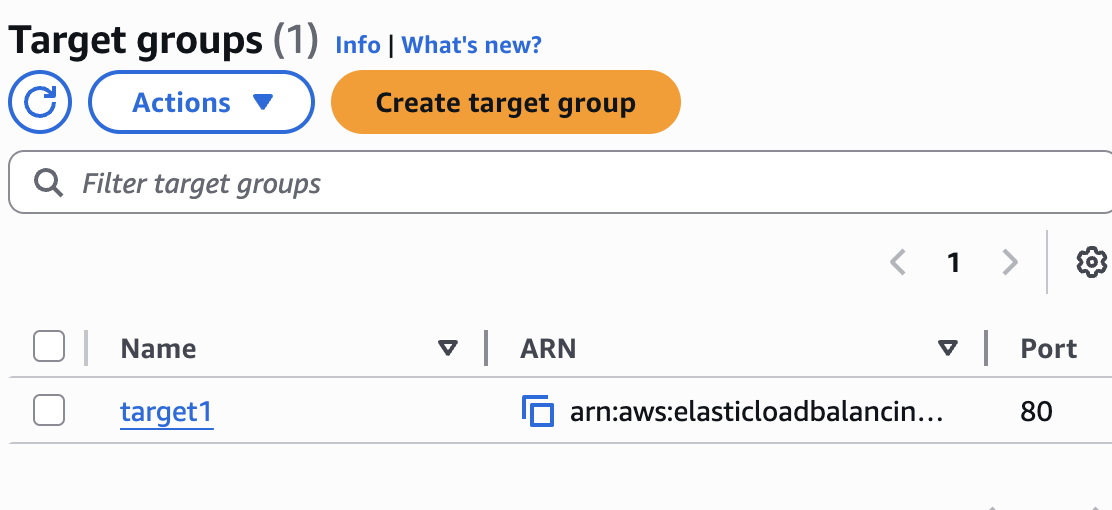


Next keep every thing by default click on **Next**

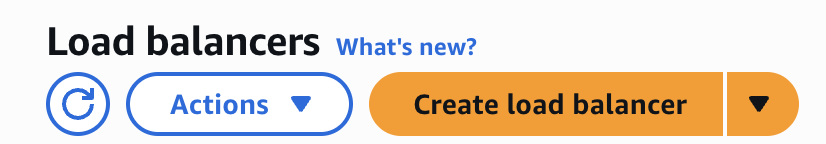
**>>select the instances for target group**

****

**>>click on create a target group**

****

Task 8**:Create Load balancer**

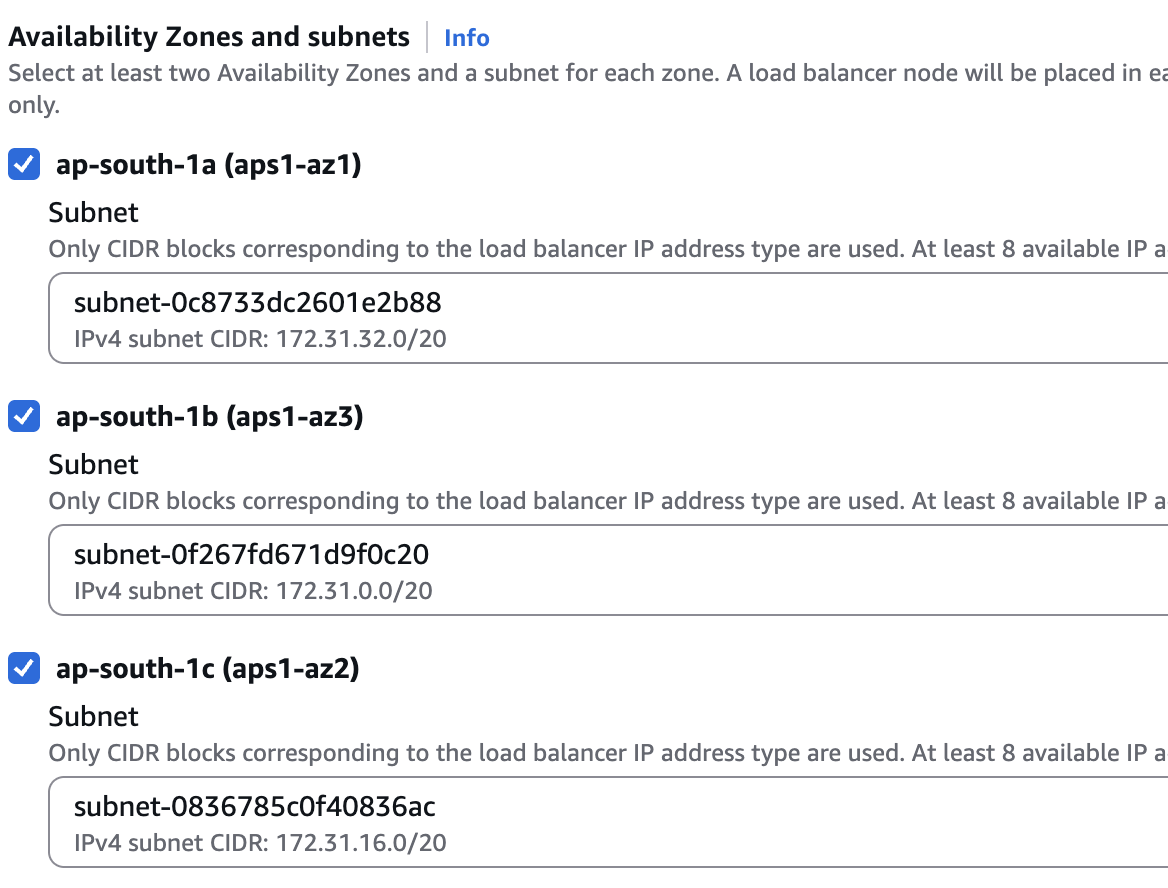
****

**>>Select:**Application load balancer

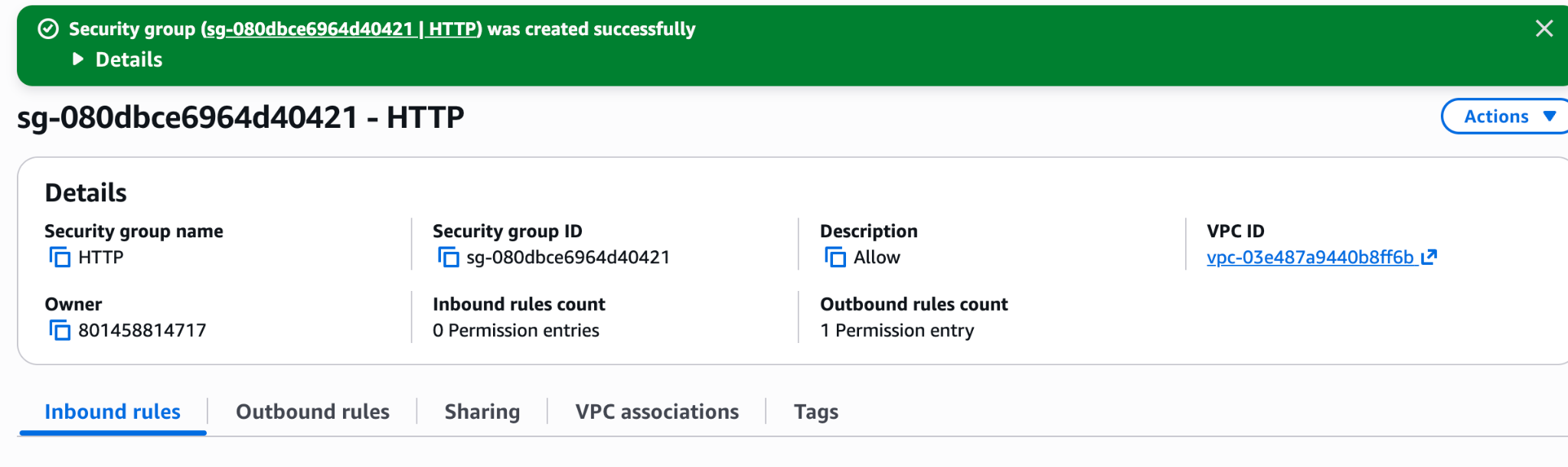
**>> configure the Application load balancer**

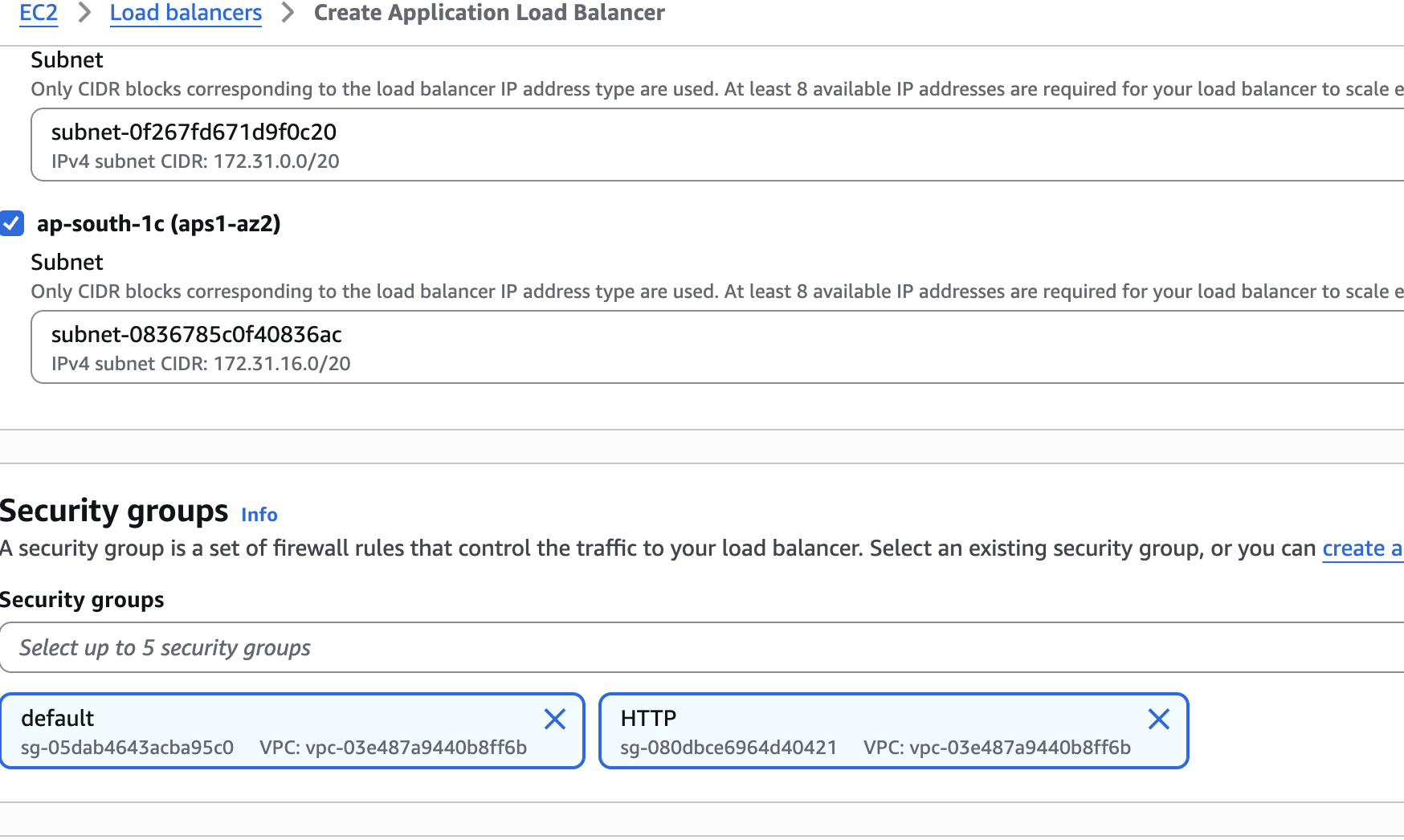
**Load balancer name :** applicationloadbalancer

>> Need to select atleast two availability zones to enroute the load balancer traffic

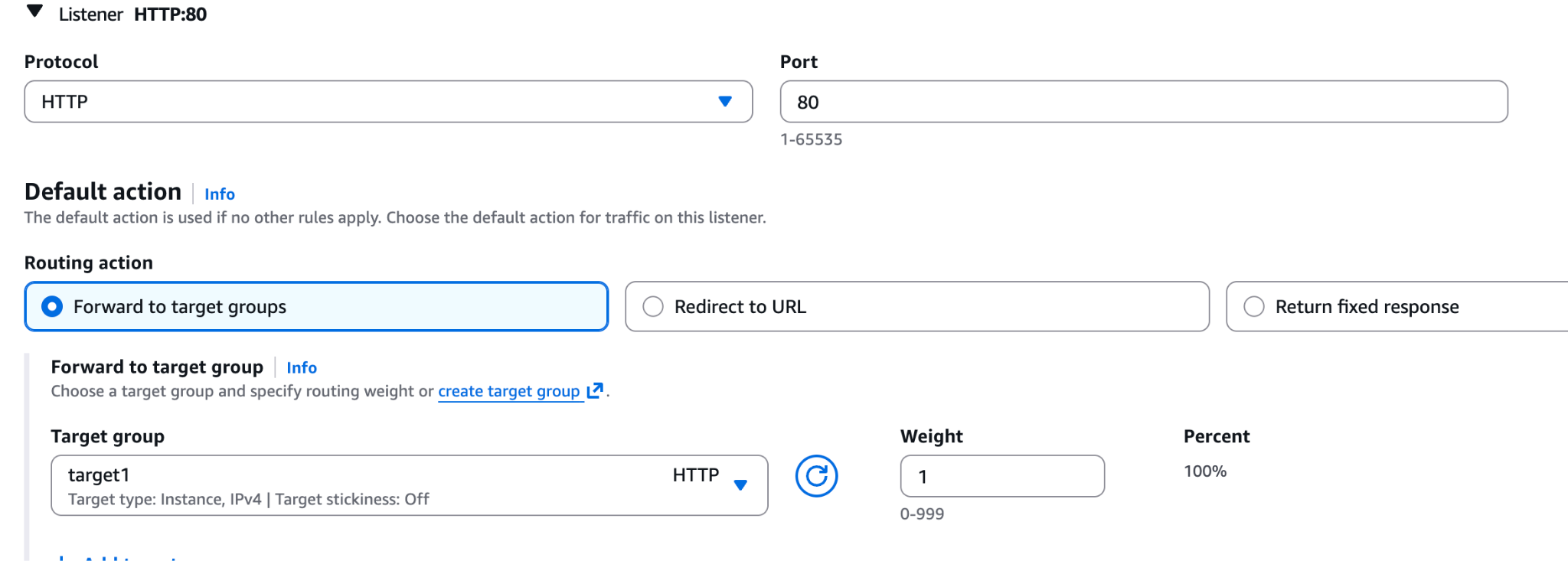


>>create security group. I created security group name it has HTTP - IPV4

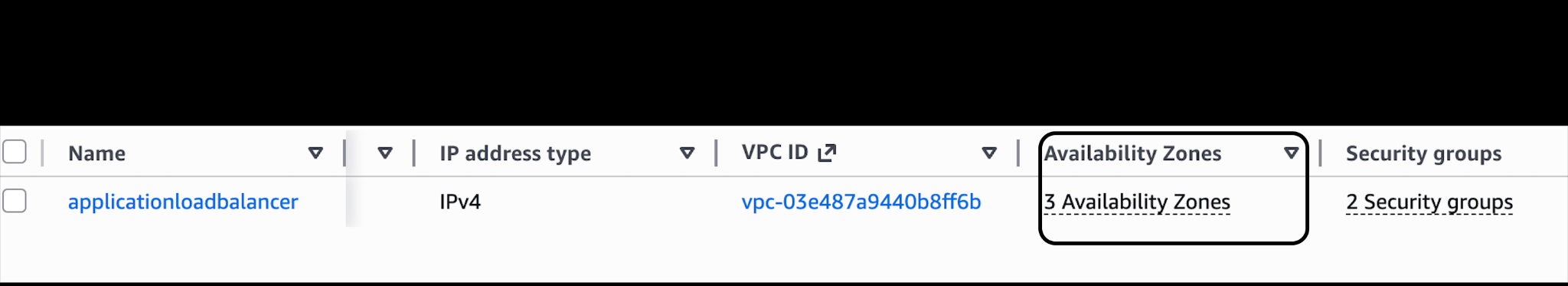


>

>> Check the HTTP PORT and select the Target group

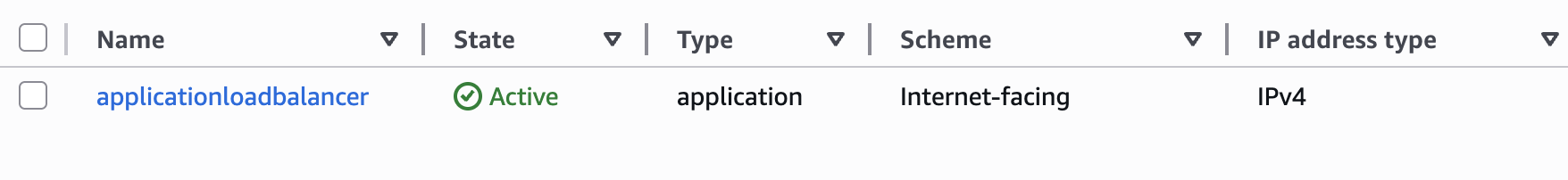


>> remaining leave every thing by default and create load balancer

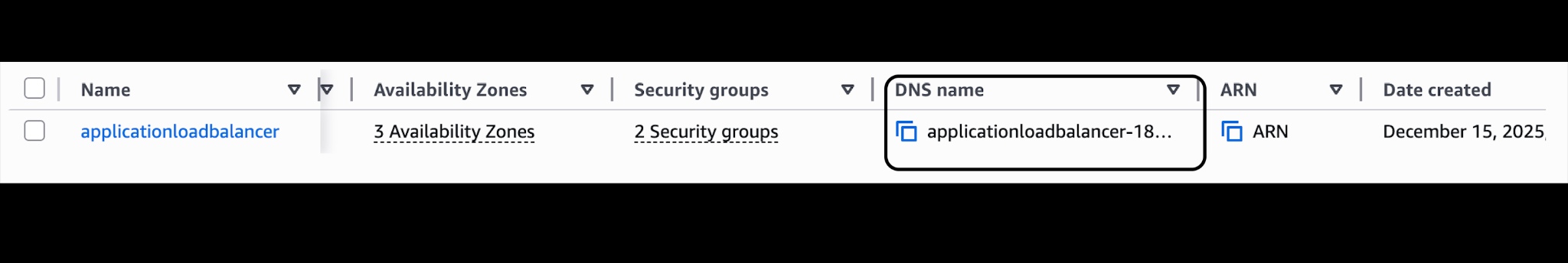


**INFO :** 3 availability zone –load balancer it will distribute the traffic in three availability zone instances

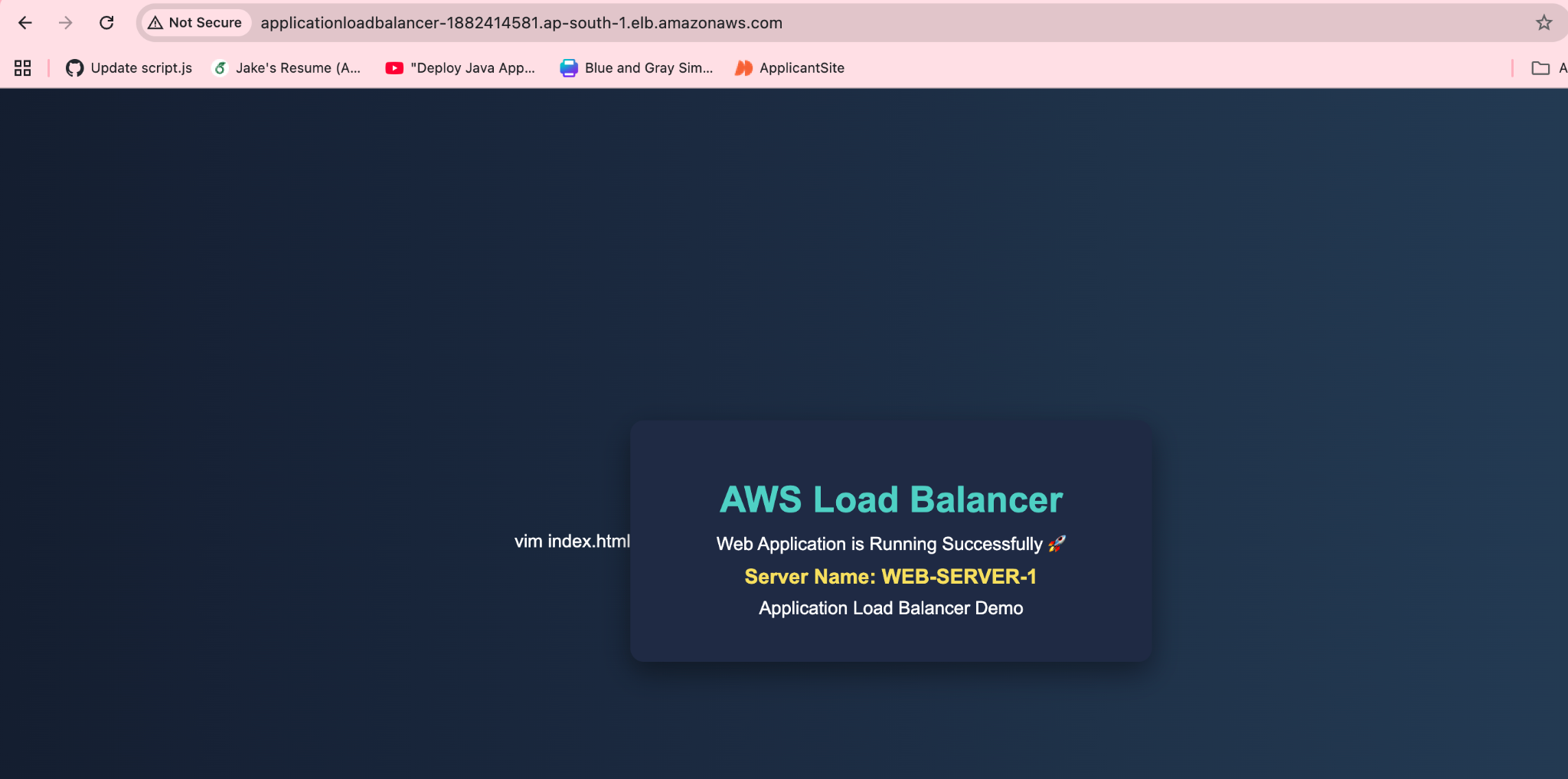
If the load state is in provisioning wait for few minutes them it will be in active



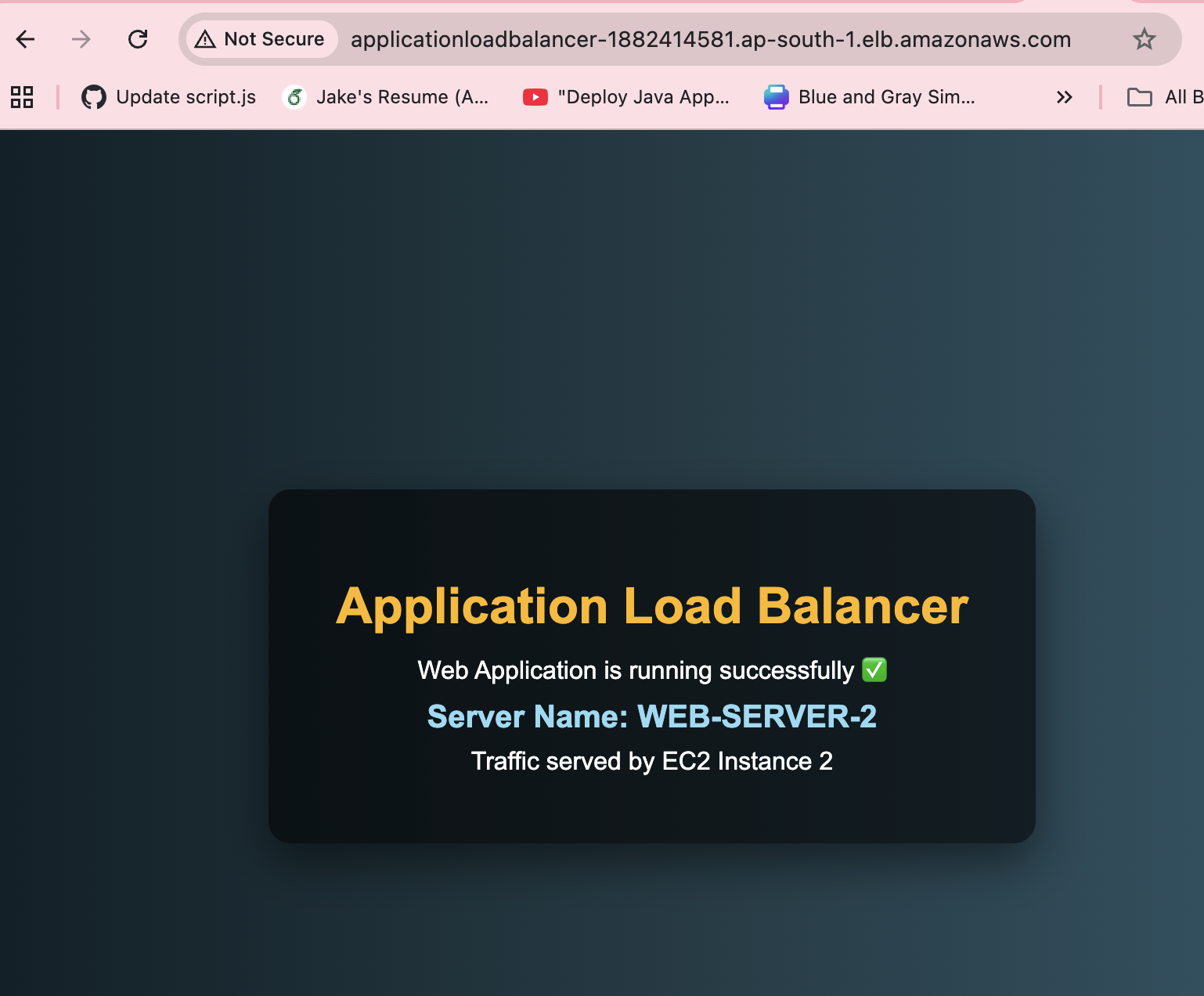
>To access the application no need to use the ip address use the DNS name for the safe and secure we can also send DNS name to the users (we are not using the ip address we only using the DNS name)



>> copy the DNS name paste it in chrome webserver-1 page will come



>> if we refresh the page webserver-2 page will come



**>>INFO →** traffic is distributing the by using the traffic group in load balancer. We can run multiple servers in traffic group. So it will distribute the traffic. When we send any request our target group will immediately take the action any server can give response to the user (like which ever server having low load the traffic group will send the request to that server)